LA-UR-19-28544
Approved for public release; distribution is unlimited.

Title: Boron effect on active measurements with the Dynamic Albedo of Neutrons on board the Mars Curiosity Rover

Author(s): Festal, Sophie
Nowicki, Suzanne Florence
Czarnecki, Sean
Gasda, Patrick J.
Hardgrove, Craig

Intended for: Report

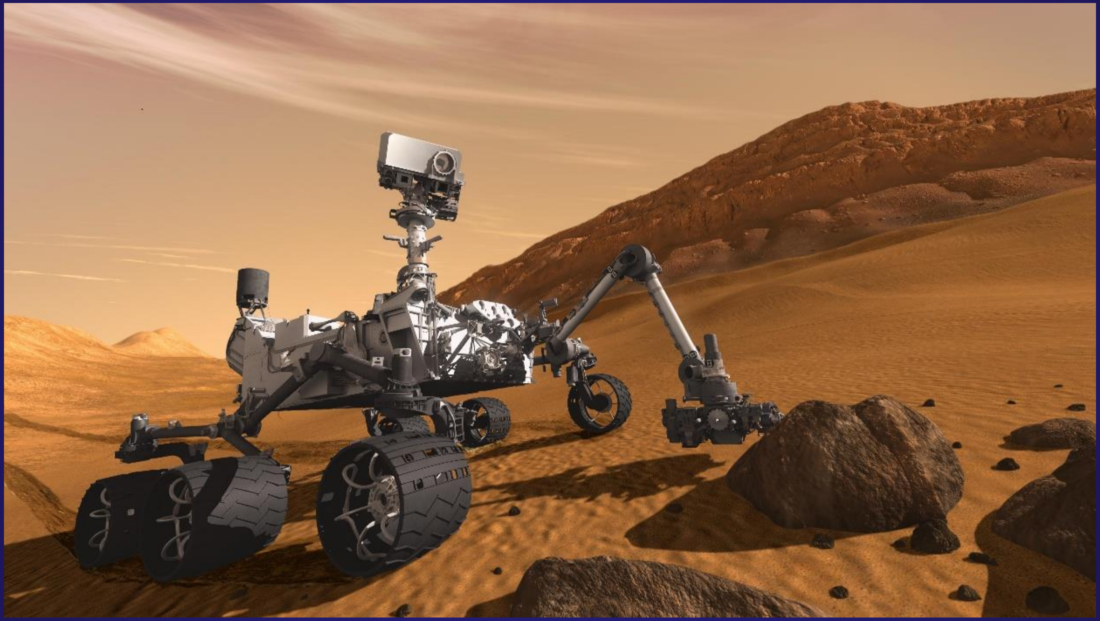
Issued: 2019-08-23

Los Alamos
NATIONAL LABORATORY

EST. 1943

NNSA
National Nuclear Security Administration

Boron effect on active measurements with the Dynamic Albedo of Neutrons on board the Mars Curiosity Rover



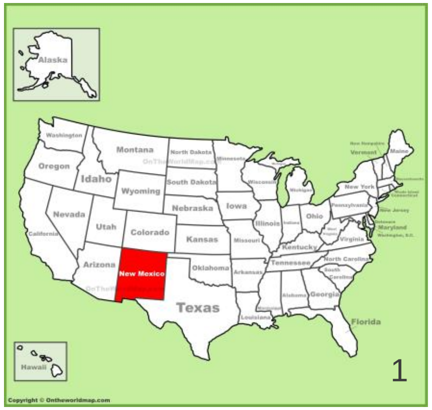
https://www.nasa.gov/sites/default/files/images/551041main_pia14156-full_full.jpg

Sophie Festal[1], Suzanne Nowicki[1], Sean Czarnecki[2], Patrick Gasda[2], Craig Hardgrove[3]

1: ISR-1: Space Science and Applications, 2: ISR-2: Space and Remote Sensing, 3: Arizona State University

New Mexico

Ghost Ranch[2]



Santa Fe Museum of art[3]

Valles Caldera[4]

Abiquiu lake[5]

1: http://ontheworldmap.com/usa/state/new-mexico/ 2: https://anncavittfisher.com/2017/05/12/ghost-ranch/ 3: https://santafe.com/article/all-things-real-estate-the-new-mexico-museum-of-art
4: https://www.nps.gov/vall/index.htm 5: https://bunnyterry.com/cool-off-abiquiu-lake-new-mexico/

Outline

- Introduction
- The DAN instrument
- Objectives
- Hydrogen detection & simulations
- Results & analyses
- Conclusion – Future work
- Annexes

Introduction

- MSL mission:
  - Probing Martian soil, analyze its composition
  - Understand the past of Mars:
    - Life?
    - How much water was there 3.5 billion years ago?

- 2002: Odyssey Spacecraft detects a signal for water
- 2012 (August 6th): Mars Curiosity Rover landed in Gale Crater
- 2013: hydrogen presence confirmed by ChemCam

- An ancient lake formed Gale crater → volcanic rocks reacted with water → water trapped in rocks today

https://www.jpl.nasa.gov/missions/mars-science-laboratory-curiosity-rover-msl/

Introduction

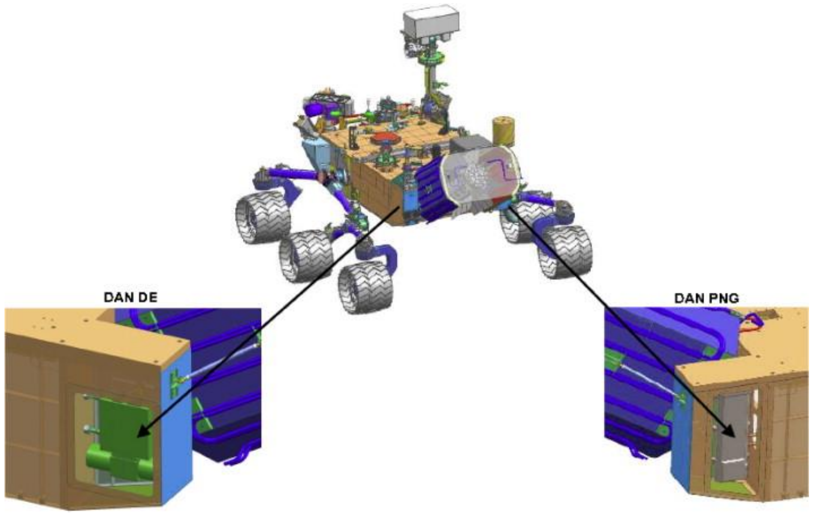
- To measure the water contained in those rocks, water content is proportional to hydrogen content
- Water content unit: wt% WEH (water equivalent hydrogen)
  - In a water molecule (H_2O), hydrogen has a certain relative mass:

$$100wt\%\, WEH = \frac{2\times H}{(2\times H+O)} \approx 0.11u$$

  - So 1wt% WEH (=0.0011u) is the hydrogen mass that corresponds to 1% of the total hydrogen mass in a water molecule.
- In Martian soils in Gale Crater, average WEH = 2-4 wt%

https://www.jpl.nasa.gov/missions/mars-science-laboratory-curiosity-rover-msl/

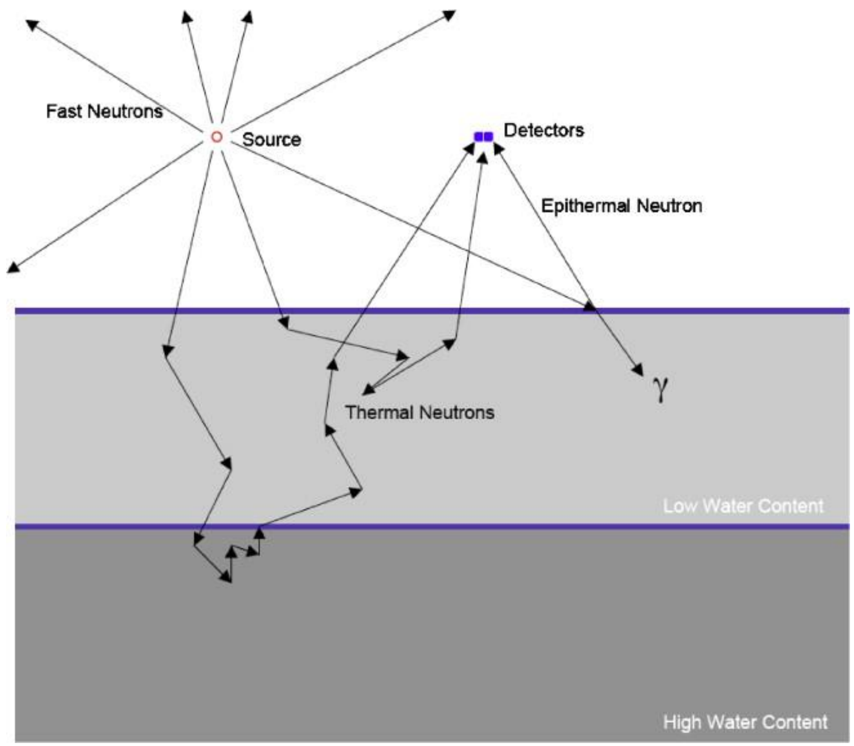
The Dynamic Albedo of Neutrons (DAN) instrument



2 He-3 tubes:
- Bare: thermal + epithermal
- Covered in Cd: epithermal
- Difference is a measure of thermal neutrons
- $^{3}He + n \rightarrow {}^{1}H + {}^{3}H + 0.764MeV$

- Pulsed neutron generator
- 14.1MeV neutrons
- 10^7 neutrons/pulse
- 10 pulses/s

How does DAN works?



Objectives

The primary objective of the DAN instrument is to quantify the amount of H in the soil of Mars.

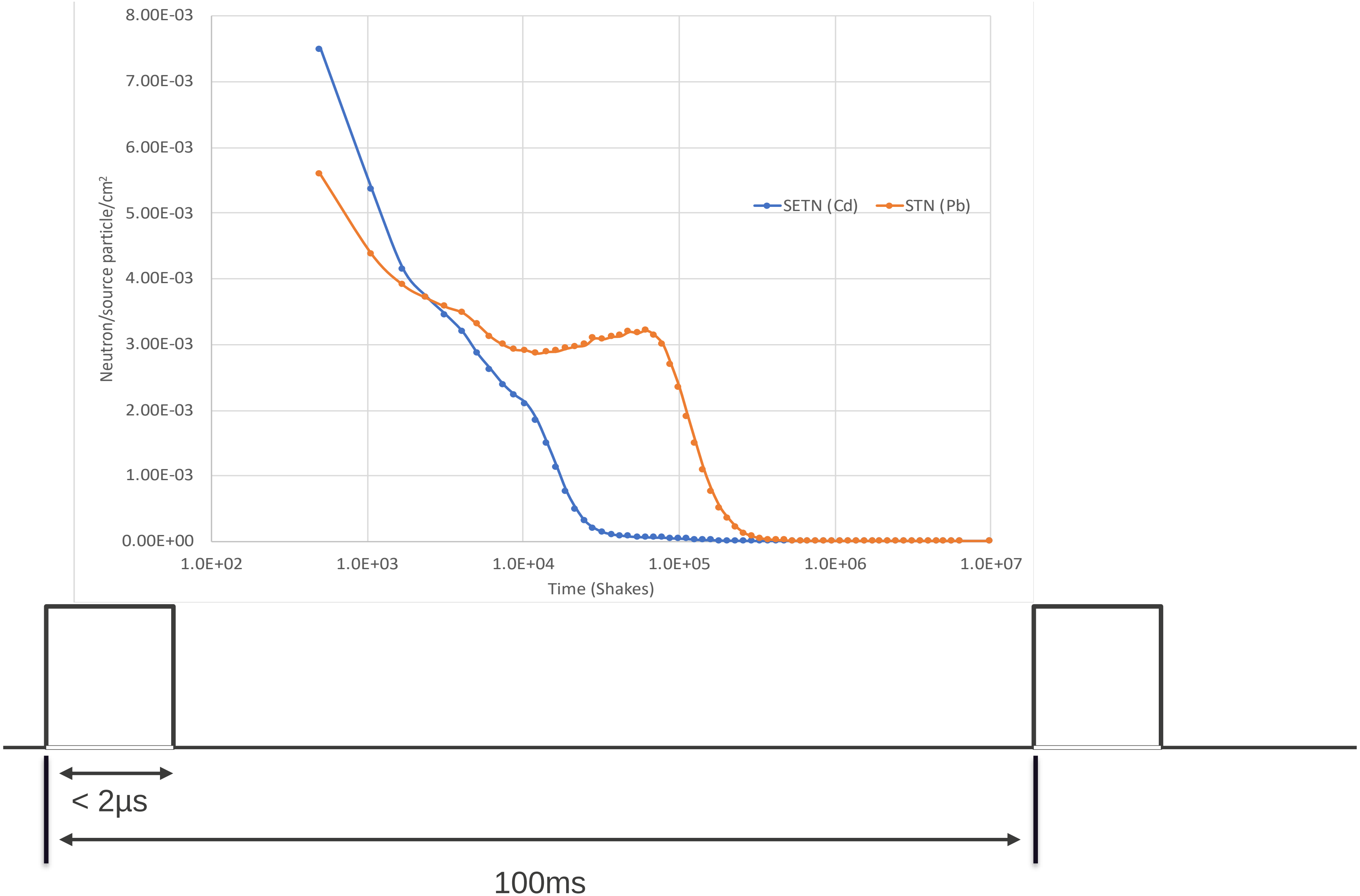
Because H is a light element, it is a really good moderator for neutrons. By measuring thermal neutrons, one can quantify H. However, other elements, such as B, have a high cross section for thermal neutron capture, and can affect the thermal neutron flux.

Recently, ChemCam has detected between 100 and 500ppm B in veins, in the Murray formation and Yellowknife Bay (Gasda et al., 2017). We would like to understand how the concentration of B affect the thermal neutrons detected with DAN, and quantify the uncertainty of the DAN measurement in the presence of B.

Tasks

- Modify MCNP6 input files, implement the right soil composition
  - The soil compositions come from ChemCam data (sebina)
  - Hydrogen: 0 → 6 wt% WEH
  - Boron: 0 → 300ppm
- Run multiple files using computer cluster (mpi)
- Analyze the results:
  - take the difference between bare and Cd-wrapped det to calculate thermal neutron count rate,
  - integrate,
  - calculate the uncertainties over time,
  - make 3D plot for a typical DAN measurements.
  - Analysis tool: ROOT

Time Structure of the DAN Instrument



Results & analyses – The neutron die-way technique

Analyzing the data: the neutron die-away technique.

- Fast neutrons interacting with Martian soils will be thermalized in different ways depending on the target's elemental composition.
- If the hydrogen content increases, thermal neutron arrival times shift to earlier times.
- Binning by time interval the thermal neutron counts allows to see these differences in arrival time and curve shapes.

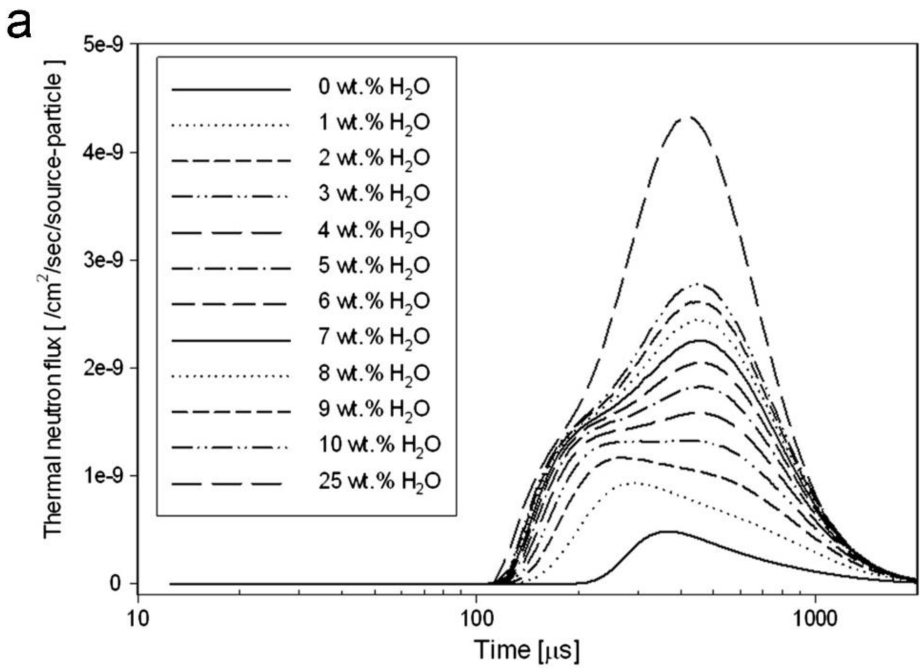
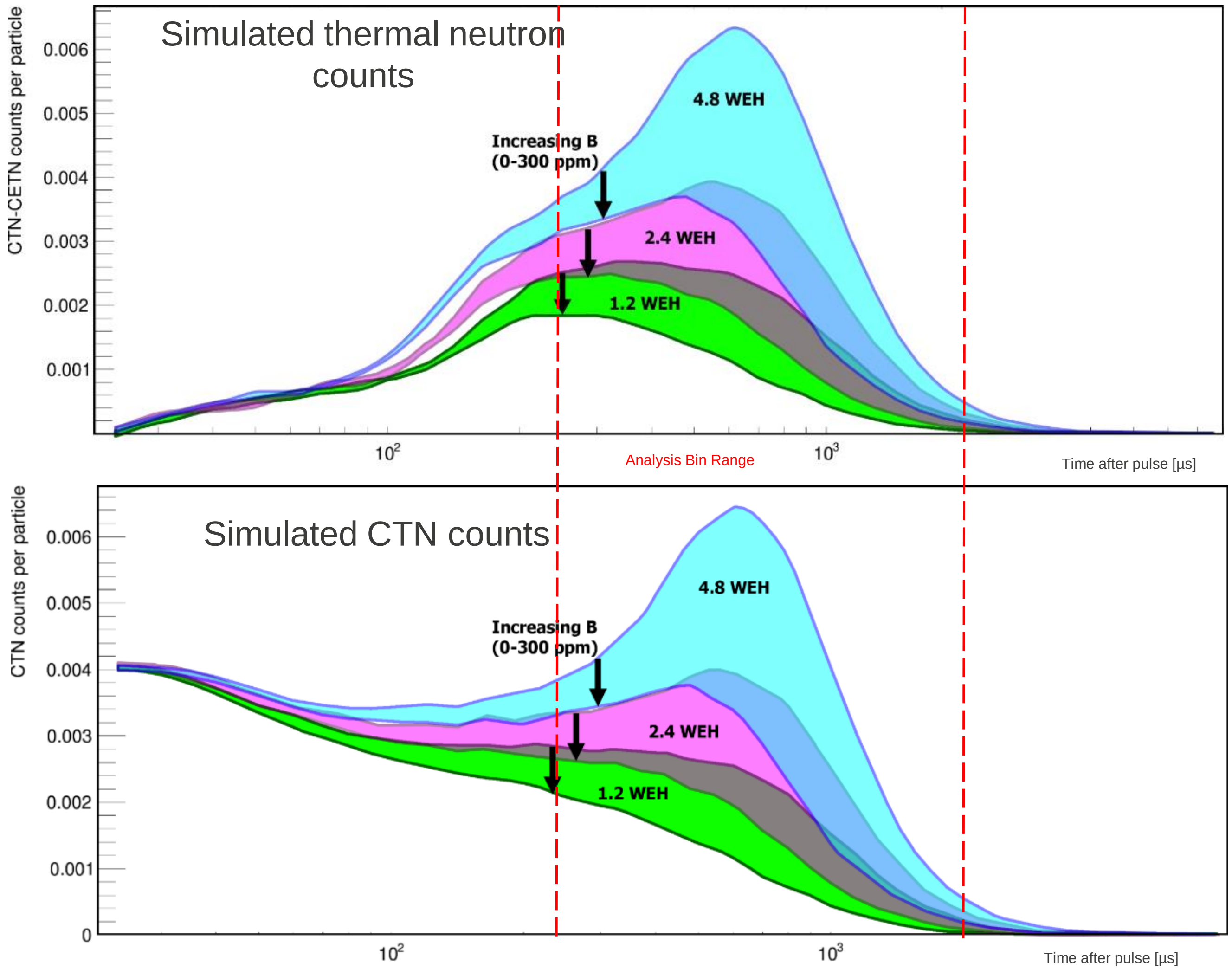


Fig. Example of neutron flux at the ^{3}He neutron detector versus time of arrival ("die-away" curve) for thermal neutrons.

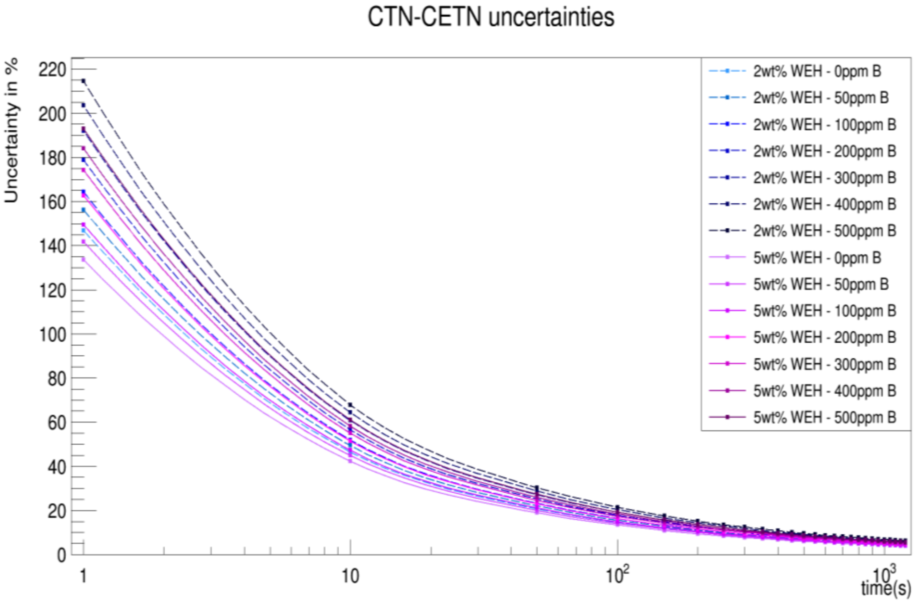
(C. Hardgrove et al., 2011)

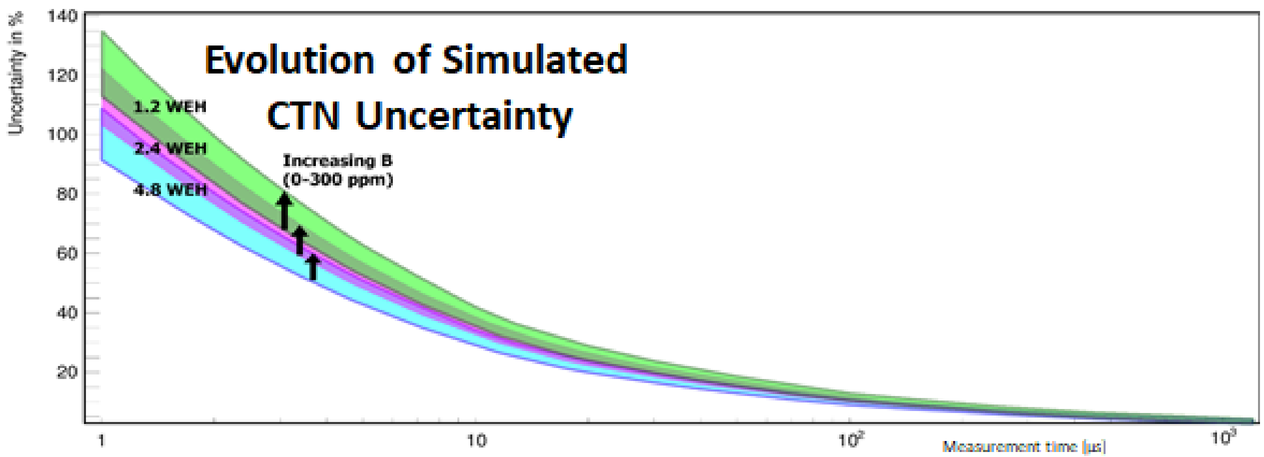
Results & analyses – Some results



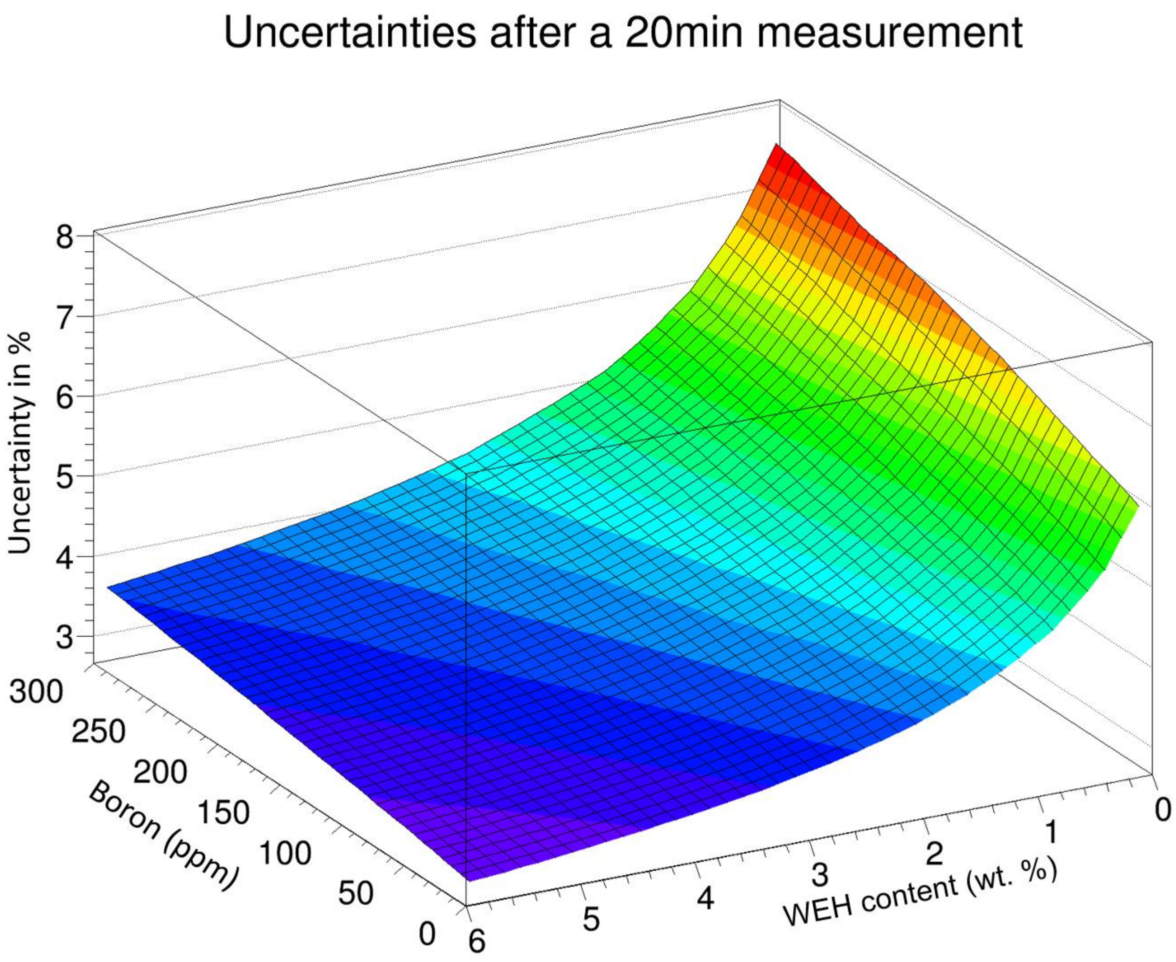
Uncertainties

- The uncertainties are calculated by:
  - Summing the number of counts per pulse
  - Multiplying by the number of pulses per second
  - Integrating the number of counts per second N over time
  - $\sqrt{N}/N$ is the uncertainty for N counts





Uncertainties – 3D plot



- 3D plot using ROOT by CERN
- When B amount increases, uncertainties increase
- When H content decreases, uncertainties increase

Conclusion - Future Work

- The bulk water content is measured by detecting thermal neutrons thermalized by hydrogen and analyzing their die-away curves.
- Other elements present in Mars's subsurface, such as boron, absorb thermal neutrons and have an effect on the measurements, therefore on their uncertainties.
- Low hydrogen content + high boron amount increase the uncertainties but stay below 7.5% for 0wt% WEH and 300ppm boron.

- Future work:
  - Same analysis with high absorbers such as chlorine and iron
  - Different geometries: model of veins

Thank you !

Annexes

Annexes – Input files

- The soil composition in the MCNP6 input files for 2.4wt% WEH and 0ppm B

```
m7010
          24050       -0.000088904
          17035       -0.008030560
          26057       -0.003013218
          20048       -0.000094996
          35079       -0.000052557
          16036       -0.000003649
          28058       -0.000694956
          26054       -0.008311590
          26056       -0.130474188
          14030       -0.006652305
          20046       -0.000002034
          28062       -0.000037105
          19040       -0.000000807
          19039       -0.006405939
          22048       -0.004846451
          22050       -0.000340545
          30066.80c       -0.000225023
          12025       -0.002555551
          15031       -0.002653936
          20044       -0.001059686
          35081       -0.000051122
          14028       -0.198413842
          22049       -0.000355658
          13027       -0.045786593
          30070.80c       -0.000004955
          24053       -0.000194408
          16033       -0.000273393
          26058       -0.000401004
          22047       -0.000489114
          24054       -0.000048390
          20040       -0.049246003
          30067.80c       -0.000032788
          24052       -0.001714517
          11023       -0.014866321
          19041       -0.000462297
          17037       -0.002569440
          1001       -0.002666667
          25055       -0.001235312
          30068.80c       -0.000149716
          30064.80c       -0.000399004
          16032       -0.034625540
          14029       -0.010079575
          28064       -0.000009451
          28061       -0.000011634
          22046       -0.000542369
          12026       -0.002813661
          8016       -0.434612773
          12024       -0.020186315
          20042       -0.000328671
          28060       -0.000267690
          20043       -0.000068577
          16034       -0.001549199
        nlib=70c
```

Annexes – ROOT program – Multigraph.C 1/2

```
{

gROOT->ProcessLine(".L Read.C");
gROOT->ProcessLine(".L makegraph_probabilities.C");
gROOT->ProcessLine(".L makegraph_uncertainties.C");
// gROOT->ProcessLine(".L file_name.C");

//TString file1,file2,file3,file4,file5,file6,file7;
Int_t nb_files=18;   //nb of input files
TString file, file_buffer;
TString g_name;
 Int_t i=0;
 Int_t count1=3; //nb of groups (0.0H, 0.4H...)
 Int_t count2=6; //nb of lines in a group



 TString file1="1.2H_0.901BNACS_14.22Fe_1.06Cl_homog.o";
 TString file2="1.2H_0.947BNACS_14.22Fe_1.06Cl_60B_homog.o";
 TString file3="1.2H_0.993BNACS_14.22Fe_1.06Cl_120B_homog.o";
 TString file4="1.2H_1.039BNACS_14.22Fe_1.06Cl_180B_homog.o";
 TString file5="1.2H_1.085BNACS_14.22Fe_1.06Cl_240B_homog.o";
 TString file6="1.2H_1.131BNACS_14.22Fe_1.06Cl_300B_homog.o";

 TString file7="2.4H_0.901BNACS_14.22Fe_1.06Cl_homog.o";
 TString file8="2.4H_0.947BNACS_14.22Fe_1.06Cl_60B_homog.o";
 TString file9="2.4H_0.993BNACS_14.22Fe_1.06Cl_120B_homog.o";
 TString file10="2.4H_1.039BNACS_14.22Fe_1.06Cl_180B_homog.o";
 TString file11="2.4H_1.085BNACS_14.22Fe_1.06Cl_240B_homog.o";
 TString file12="2.4H_1.131BNACS_14.22Fe_1.06Cl_300B_homog.o";

 TString file13="4.8H_0.9BNACS_14.22Fe_1.06Cl_homog.o";
 TString file14="4.8H_0.946BNACS_14.22Fe_1.06Cl_60B_homog.o";
 TString file15="4.8H_0.992BNACS_14.22Fe_1.06Cl_120B_homog.o";
 TString file16="4.8H_1.038BNACS_14.22Fe_1.06Cl_180B_homog.o";
 TString file17="4.8H_1.084BNACS_14.22Fe_1.06Cl_240B_homog.o";
 TString file18="4.8H_1.13BNACS_14.22Fe_1.06Cl_300B_homog.o";



 TCanvas *cv = new TCanvas("cv","cv",2400,1200); TMultiGraph *mg = new TMultiGraph();
 auto legend = new TLegend(0.7,0.3,0.9,0.9);
```

```
 std::vector<TString> array_files;
 array_files.push_back(file1);
array_files.push_back(file2);
array_files.push_back(file3);
array_files.push_back(file4);
array_files.push_back(file5);
array_files.push_back(file6);
array_files.push_back(file7);
array_files.push_back(file8);
array_files.push_back(file9);
array_files.push_back(file10);
array_files.push_back(file11);
array_files.push_back(file12);
array_files.push_back(file13);
array_files.push_back(file14);
array_files.push_back(file15);
array_files.push_back(file16);
array_files.push_back(file17);
array_files.push_back(file18);
```

```
for (Int_t j=0; j<count1; j++)
  {

    for (Int_t k=0; k<count2; k++)
      {
       file_buffer=array_files[i];
       Double_t **array14;  array14=Read(file_buffer,14);
       Double_t **array24;  array24=Read(file_buffer,24);

       TGraph *g; g=makegraph_probabilities(file_buffer,array14,array24);
       mg->Add(g);
```

Annexes – ROOT program – Multigraph.C 2/2

```
if (j==0)
          {

            g->SetLineStyle(9); g->SetLineWidth(2); g_u->SetLineStyle(9); g_u->SetLineWidth(2); g_name="1.2wt% WEH - ";

          if (k==0){ g->SetLineColor(kAzure+1);  g->SetMarkerColor(kAzure+1);      g_name += "0ppm B"; }
          if (k==1){ g->SetLineColor(kAzure+2);  g->SetMarkerColor(kAzure+2);     g_name += "60ppm B"; }
          if (k==2){ g->SetLineColor(kBlue);      g->SetMarkerColor(kBlue);           g_name += "120ppm B"; }
          if (k==3){ g->SetLineColor(kBlue+1);    g->SetMarkerColor(kBlue+1);         g_name += "180ppm B"; }
          if (k==4){ g->SetLineColor(kBlue+2);     g->SetMarkerColor(kBlue+2);           g_name += "240ppm B"; }
          if (k==5){ g->SetLineColor(kBlue+3);     g->SetMarkerColor(kBlue+3);            g_name += "300ppm B"; }
        }

        if (j==1)
          {

          g->SetLineStyle(1); g->SetLineWidth(2); g_u->SetLineStyle(1); g_u->SetLineWidth(2); g_name="2.4wt% WEH - ";

          if (k==0){ g->SetLineColor(kViolet-4);  g->SetMarkerColor(kViolet-4);     g_name += "0ppm B"; }
          if (k==1){ g->SetLineColor(kViolet-2);  g->SetMarkerColor(kViolet-2);      g_name += "60ppm B"; }
          if (k==2){ g->SetLineColor(kViolet);      g->SetMarkerColor(kViolet);          g_name += "120ppm B"; }
          if (k==3){ g->SetLineColor(kMagenta);    g->SetMarkerColor(kMagenta);       g_name += "180ppm B"; }
          if (k==4){ g->SetLineColor(kMagenta+1);     g->SetMarkerColor(kMagenta+1);          g_name += "240ppm B"; }
          if (k==5){ g->SetLineColor(kMagenta+2);     g->SetMarkerColor(kMagenta+2);         g_name += "300ppm B"; }
        }

        if (j==2)
          {
            g->SetLineStyle(9); g->SetLineWidth(2); g_u->SetLineStyle(9); g_u->SetLineWidth(2); g_name="4.8wt% WEH - ";

            if (k==0){ g->SetLineColor(kSpring+5);  g->SetMarkerColor(kSpring+5);    g_name += "0ppm B"; }
          if (k==1){ g->SetLineColor(kSpring-5);  g->SetMarkerColor(kSpring-5);     g_name += "60ppm B"; }
          if (k==2){ g->SetLineColor(kSpring-1);      g->SetMarkerColor(kSpring-1);        g_name += "120ppm B"; }
          if (k==3){ g->SetLineColor(kSpring-6);    g->SetMarkerColor(kSpring-6);      g_name += "180ppm B"; }
          if (k==4){ g->SetLineColor(kGreen+2);     g->SetMarkerColor(kGreen+2);          g_name += "240ppm B"; }
          if (k==5){ g->SetLineColor(kGreen+3);     g->SetMarkerColor(kGreen+3);     g_name += "300ppm B"; }
          }


        g->SetTitle(g_name);   g->SetName(g_name); legend->AddEntry(g,g_name,"lp");
        i=i+1;
       }
   }

  gPad->SetLogx();

  mg->SetTitle("Evolution of the thermal neutrons counts;time(us);CTN-CETN counts per particle");
  mg->Draw("APC");
  legend->Draw();

  return;


}
```

Annexes – ROOT program – Plotting counts

```
// function that will make 2 TGraphs for 1 file: one for the thermal neutron probabibilities and another one for uncertain
ies

TGraph* makegraph_probabilities (TString file, Double_t **array14, Double_t **array24){



  Int_t nb_lines;
  Int_t count=42;  // nb of lines in the arrays
  Int_t count2=26;  //nb of lines for uncertainties (1s, 10s, 50s, 100s, 150s, 200s, ... ,1150s, 1200s) and 1200s = 20min,
our integration time


 //creating a 1 dimension array for the difference of counts between tallies 24 and 14
 Double_t *time; time = new Double_t[count];
 Double_t *array_diff; array_diff = new Double_t[count];
 Float_t *err; err = new Float_t[count];

 for (Int_t i=0; i<count; i++)
   {
     time[i]=array24[0][i];                                              // change tallies
     array_diff[i]=array24[1][i]-array14[1][i];
     err[i]=TMath::Sqrt(array24[2][i]*array24[2][i] + array14[2][i]*array14[2][i]);
     void_array[i]=0;

   }


 TGraph* g = new TGraph (count, time, array_diff);
 // TGraphErrors* g = new TGraphErrors (count, time, array_diff,void_array,err);


 g->SetTitle("CTN-CETN curve - 50ppm boron - 1wt% WEH;time(shakes);counts per 14MeV neutron");   // change boron amount
 g->SetMarkerStyle(7);

  return g;

}
```

Annexes – ROOT program – Plotting uncertainties 1/2

```
// function that will make a TGraphs (a graph of uncertainties) for 1 file: one for the thermal neutron probabibilities and another one for uncertainties

TGraph* makegraph_uncertainties (TString file, Double_t **array14, Double_t **array24){


  Int_t nb_lines;
  Int_t count=42;  // nb of lines in the arrays
  Int_t count2=26;  //nb of lines for uncertainties (1s, 10s, 50s, 100s, 150s, 200s, ... ,1150s, 1200s) and 1200s = 20min, our integration time


 //creating a 1 dimension array for the difference of counts between tallies 24 and 14
 Double_t *time; time = new Double_t[count];
 Double_t *array_diff; array_diff = new Double_t[count];
 Double_t sum_probabilities=0;

 for (Int_t i=0; i<count; i++)
   {
     time[i]=array24[0][i];                                               // change tallies
     array_diff[i]=array24[1][i]-array14[1][i];
     sum_probabilities = sum_probabilities + array_diff[i];     //sum of the probrabilities
   }


  // creating a 1 dimension array for the uncertainties of thermal neutron probabilitites
  Float_t *time2; time2 = new Float_t[count2];
  Double_t *counts; counts = new Double_t[count2];
  Float_t *uncertainties_percent; uncertainties_percent = new Float_t[count2];

  Double_t sum_counts=0;
  sum_counts = sum_probabilities * 10;  // Sum_counts is the number of neutrons per sec. The PNG rate is 1e7 neutrons per pulse, 10 pulses per second, so 1e8 neutrons per second. In the input files, on the sdef card is put a weight card: wgt=5e6, which corresponds to the number of neutrons per pulse when the degeration is taken into account. To have the number of neutrons per second I only need to multiply the sum of the probability by 10, the number of pulses.

  // cout << "le nombre de neutrons par sec est: " << sum_counts << endl;

  Int_t j=50;
  time2[0]=1; counts[0]=sum_counts; uncertainties_percent[0]=((TMath::Sqrt(counts[0]))/ counts[0])*100 ;
  time2[1]=10; counts[1]=counts[0]*time2[1]; uncertainties_percent[1]=((TMath::Sqrt(counts[1]))/ counts[1])*100;

  cout << time2[0] << " " << uncertainties_percent[0] << " " << counts[0] << endl;
  cout << time2[1] << " " << uncertainties_percent[1] << " " << counts[1] << endl;
```

Annexes – ROOT program – Plotting uncertainties 2/2

```
  for (Int_t k=2; k<count2; k++)
    {
      time2[k]=j;
      counts[k]=counts[0]*time2[k];
      uncertainties_percent[k] =TMath::Sqrt(counts[k]) ;
      uncertainties_percent[k] = (uncertainties_percent[k])/counts[k];
      uncertainties_percent[k] = uncertainties_percent[k]*100;
      j=j+50;

      // cout << time2[k] << " " << uncertainties_percent[k] << " " << counts[k] << endl;
    }



  //Plotting the uncertainties related to thermal neutron probabilities as a function of time
  TGraph* g2 = new TGraph (count2, time2, uncertainties_percent);

  g2->SetTitle("Sqrt(N)/N for CTN-CETN curve - 50ppm boron - 1wt% WEH;time(s);Uncertainty in %");   // change boron amount
  g2->SetMarkerStyle(7);

  return g2;

}
```

Annexes – ROOT program – Uncertainties in 3D 1/2

```
// graph 3D avec tous les fichiers

{

  gROOT->ProcessLine(".L Read.C");
  gROOT->ProcessLine(".L uncertainties_3D.C");
 // gROOT->ProcessLine(".L file_name.C")


 Int_t nb_files=256;   //nb of input files
 TString file, file_buffer;
  Int_t i=0;
  Int_t count1=16; //nb of groups (0.0H, 0.4H...)
  Int_t count2=16; //nb of files per group

  //0.0H, 16 files
TString file1="o.0H_0.902BNACS_14.22Fe_1.06Cl_homog.mx.o";
TString file2="o.0H_0.917BNACS_14.22Fe_1.06Cl_20B_homog.mx.o";
TString file3="o.0H_0.932BNACS_14.22Fe_1.06Cl_40B_homog.mx.o";
TString file4="o.0H_0.948BNACS_14.22Fe_1.06Cl_60B_homog.mx.o";
TString file5="o.0H_0.963BNACS_14.22Fe_1.06Cl_80B_homog.mx.o";
TString file6="o.0H_0.978BNACS_14.22Fe_1.06Cl_100B_homog.mx.o";
TString file7="o.0H_0.994BNACS_14.22Fe_1.06Cl_120B_homog.mx.o";
TString file8="o.0H_1.009BNACS_14.22Fe_1.06Cl_140B_homog.mx.o";
TString file9="o.0H_1.024BNACS_14.22Fe_1.06Cl_160B_homog.mx.o";
TString file10="o.0H_1.04BNACS_14.22Fe_1.06Cl_180B_homog.mx.o";
TString file11="o.0H_1.055BNACS_14.22Fe_1.06Cl_200B_homog.mx.o";
TString file12="o.0H_1.07BNACS_14.22Fe_1.06Cl_220B_homog.mx.o";
TString file13="o.0H_1.086BNACS_14.22Fe_1.06Cl_240B_homog.mx.o";
TString file14="o.0H_1.101BNACS_14.22Fe_1.06Cl_260B_homog.mx.o";
TString file15="o.0H_1.116BNACS_14.22Fe_1.06Cl_280B_homog.mx.o";
TString file16="o.0H_1.132BNACS_14.22Fe_1.06Cl_300B_homog.mx.o";
```

```
std::vector<TString> array_files;
array_files.push_back(file1);
array_files.push_back(file2);
array_files.push_back(file3);
array_files.push_back(file4);
array_files.push_back(file5);
array_files.push_back(file6);
array_files.push_back(file7);
array_files.push_back(file8);
array_files.push_back(file9);
array_files.push_back(file10);
array_files.push_back(file11);
array_files.push_back(file12);
array_files.push_back(file13);
array_files.push_back(file14);
array_files.push_back(file15);
array_files.push_back(file16);
```

Annexes – ROOT program – Uncertainties in 3D 2/2

```
  TCanvas *cv = new TCanvas("cv", "Graph 2D example", 0,0,1000,900);    TGraph2D *g = new TGraph2D();
  Double_t B,H;
  Float_t u;
  Double_t boron_amount=0;
  Float_t h_amount=0.0;
  //g->SetTitle("Uncertainties after a 20min measurement; boron (ppm); wt% WEH; uncertainty after 20min in %");
  g->SetTitle("Uncertainties after a 20min measurement");

    for (Int_t j=0; j<count1; j++){

      boron_amount=0; //reinitialise boron amount between each file group
      for (Int_t k=0; k<count2; k++){

        file_buffer=array_files[i];
        Double_t **array14; array14=Read(file_buffer,14);
        Double_t **array24; array24=Read(file_buffer,24);

        B=boron_amount;
        H=h_amount;
        u=uncertainties_20min(file_buffer,array14,array24);

        g->SetPoint(i,B,H,u);

        boron_amount=boron_amount+20;
        i=i+1;
      }
      h_amount= h_amount+0.4;
    }

  gStyle->SetPalette(1);
  g->Draw("surf1");
  return ;



}
```

Hydrogen detection & simulations

- The most abundant chemical elements present in the Martian soils that have an influence on hydrogen detection: boron, chlorine, iron, titanium.

Element	Thermal neutron absorption cross section
Boron	767 barns
Chlorine	33.5 barns
Iron	2.56 barns
Titanium	6.09 barns

- Titanium is in the rover's wheels, taken into account in the simulations